

Amendments to the Claims

The following listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (original) In a wireless communications device, a method for updating system software stored in memory, the method comprising: storing system software for the wireless device in a plurality of current code sections; receiving a new code section; arranging the new code section with current code sections to form updated system software for the wireless device; and, executing the updated system software.
2. (original) The method of claim 1 further comprising: identifying a first code section for updating; and, wherein arranging the new code section with current code sections to form updated system software includes replacing the first code section with the new code section.
3. (original) The method of claim 2 wherein executing the updated system software includes using the new code section in executing the updated system software.
4. (original) The method of claim 1 further comprising: forming the system software into a first plurality of symbol libraries, each symbol library comprising at least one symbol; and, arranging the first plurality of symbol libraries into a second plurality of code sections.
5. (original) The method of claim 4 wherein forming system software into a first plurality of symbol libraries includes each symbol library comprising symbols having related functionality.
6. (original) The method of claim 5 wherein receiving a new code section includes receiving the new code section via a wireless communications device air interface.
7. (original) The method of claim 6 wherein forming system software into a first plurality of symbol libraries includes forming read-write data for a plurality of symbol libraries; and, wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the read-write data in a shared read-write code section.
8. (original) The method of claim 7 wherein receiving a new code section includes receiving an updated read-write code section.
9. (original) The method of claim 8 wherein identifying a first code section for updating includes identifying the read-write code section; wherein arranging the new code section with current

- code sections to form updated system software includes replacing the read-write code section with the updated read-write code section; and wherein executing the updated system software includes using the updated read-write code section in executing of the updated system software.
10. (original) The method of claim 9 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes starting symbol libraries at the start of code sections; the method further comprising: storing the start of code sections at corresponding start addresses; and, maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses.
 11. (original) The method of claim 10 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging each symbol to be offset from its respective code section start address; and the method further comprising: maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers.
 12. (original) The method of claim 11 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes: arranging the code section address table in a first table code section; and, arranging the symbol offset address table in a second table code section.
 13. (original) The method of claim 12 wherein receiving an updated code section includes receiving an updated first table code section and an updated second table code section; wherein arranging the new code section with current code sections to form updated system software includes replacing the first table code section with the updated first table code section, and the second table code section with the updated second table code section; and, wherein executing the updated system software includes using the updated first table code section and updated second table code section in executing the updated system software.
 14. (original) The method of claim 13 wherein forming system software into a first plurality of symbol libraries includes forming a patch library; wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a patch library into a patch manager code section; wherein arranging the new code section with current code sections to form updated system software for the wireless device includes: accessing the patch manager code section; and, invoking the patch library to store the new code section.

15. (original) The method of claim 14 wherein invoking the patch library to store the new code section includes invoking the patch library to over-write the first code section with the new code section.
16. (original) The method of claim 15 further comprising: after receiving the new code section, storing the new code section in a memory file system section; and, wherein arranging the new code section with current code sections to form updated system software includes invoking the patch library to over-write the first code section with the new code section stored in the memory file system section.
17. (original) The method of claim 16 wherein receiving a new code section includes receiving an updated patch manager code section; wherein arranging the new code section with current code sections to form updated system software includes replacing the patch manager code section with the updated patch manager code section; and, wherein executing the updated system software includes using the updated patch manager code section in executing the updated system software.
18. (original) The method of claim 17 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging read-write data, the code section address table, and the symbol offset address table in the patch manager code section; and, wherein receiving an updated patch manager code section includes receiving an updated symbol offset address table, an updated code section address table, and updated read-write data.
19. (original) The method of claim 17 wherein forming system software into a first plurality of symbol libraries includes forming a symbol accessor code; wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the symbol accessor code in the patch manager code section; and, the method further comprising: storing the symbol accessor code address at a first location in memory; wherein executing the updated system software includes: in response to referencing the first location in memory, accessing the symbol accessor code; and, invoking the symbol accessor code to calculate the address of a sought symbol using a corresponding symbol identifier, and a corresponding code section identifier.
20. (original) The method of claim 19 wherein invoking the symbol accessor code to calculate the address of the sought symbol includes accessing the code section address table and the symbol offset address table to calculate the address of the sought symbol.

21. (original) The method of claim 20 wherein receiving an updated patch manager code section includes receiving an updated symbol accessor code; wherein replacing the patch manager code section with updated patch manager code section includes replacing the symbol accessor code with updated symbol accessor code; and, wherein executing the updated system software includes using the updated symbol accessor code in executing the updated system software.
22. (original) The method of claim 21 further comprising: receiving an updated symbol accessor code address; storing the updated symbol accessor code address in the file system section; replacing the symbol accessor code address in the first location in memory with updated symbol accessor code address from the file system section; and, wherein executing the updated system software includes using the updated symbol accessor code address in executing the updated system software.
23. (original) The method of claim 22 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging the symbol accessor code address in the patch manager code section; and, wherein replacing the symbol accessor code address in the first location in memory with the updated symbol accessor code address from the file system section includes replacing the symbol accessor code address in the patch manager code section with the updated symbol accessor code address in an updated patch manager code section.
24. (original) The method of claim 23 wherein executing the system software includes: loading the read-write data, the code section address table, the symbol offset address table, the patch library, symbol accessor code, and a symbol accessor code address from the patch manager code section into read-write volatile memory; and, accessing the read-write data, the code section address table, the symbol offset address table, patch library, the symbol accessor code, and the symbol accessor code address from read-write volatile memory.
25. (original) The method of claim 24 wherein storing the start of code sections at corresponding start addresses includes: creating a second plurality of contiguously addressed memory blocks; identifying each memory block with a corresponding code section; and, storing code sections in the identified memory blocks.
26. (original) The method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a third plurality of symbol libraries in a first code section; wherein identifying each memory block with a corresponding code section includes

identifying a first memory block with the first code section; wherein storing code sections in the identified memory blocks includes storing the third plurality of symbol libraries in the first memory block; wherein receiving a new code section includes receiving an updated first code section with third plurality of symbol libraries arranged within; and, wherein arranging the new code section with current code sections to form updated system software for the wireless device includes overwriting the first code section in the first memory block with an updated first code section.

27. (original) The method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging a first symbol library in a first code section; wherein identifying each memory block with a corresponding code section includes identifying a first memory block with the first code section; wherein storing code sections in the identified memory blocks includes storing the first symbol library in the first memory block; wherein receiving a new code section includes receiving an updated first code section with first symbol library arranged within; and, wherein arranging the new code section with current code sections to form updated system software for the wireless device includes overwriting the first code section in the first memory block with an updated first code section.
28. (original) The method of claim 25 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate arranged symbol libraries; and, wherein creating a second plurality of contiguously addressed memory blocks includes sizing memory blocks to accommodate corresponding code sections.
29. (original) The method of claim 28 wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate sizes larger than the arranged symbol libraries.
30. (original) The method of claim 4 wherein storing system software for the wireless device in a plurality of current code sections includes storing system software in a second plurality of code sections; wherein receiving a new code section includes receiving a second plurality of updated code sections; wherein arranging the new code section with the current code section includes replacing the second plurality of current code sections with the second plurality of updated code sections; and, wherein executing the updated system software includes using the second plurality of updated code sections in executing the updated system software.

31. (currently amended) In a wireless communications device, a method for updating system software stored in memory, the method comprising: storing system software for the wireless device in a plurality of current code sections; receiving new code sections via a wireless communications device air interface; storing the new code sections in a memory file system section; identifying a subset of the plurality of current code sections for updating; replacing the identified subset of the plurality of current code sections with new code sections to form updated system software for the wireless device; and, executing the updated system software.
32. (original) The method of claim 31 further comprising: forming the system software into a first plurality of symbol libraries including a code section address table, a symbol offset address table, a symbol accessor code, a patch library, and read-write data for a plurality of symbol libraries; arranging the code section address table, the symbol offset address table, the symbol accessor code, the read-write data, the patch library, and the symbol accessor code address into a patch manager code section; wherein receiving new code sections includes receiving a new patch manager code section; and, wherein replacing current code sections with new code sections to form updated system software for the wireless device includes replacing a current patch manger code section with the new patch manager code section.
33. (original) In a wireless communications device, a software updating system, the system comprising: a code storage section memory including executable wireless device system software differentiated into a plurality of current code sections; a file system section memory for receiving new code sections; a patch library to arrange new code sections in the code storage section with the current code sections; and, wherein the arrangement of new code sections with current code sections in the code storage section forms updated executable system software.
34. (original) The system of claim 33 further comprising: wherein the file system section receives a first patch manager instruction set (PMRTI) including instruction for arranging the new code section with the current code sections; and wherein the patch library replaces the first code section in the code storage section with the new code section in response to the first PMRTI.
35. (original) The system of claim 34 wherein the code storage section comprises a first plurality of symbol libraries, each symbol library comprising at least one symbol, the first plurality of symbol libraries arranged in a second plurality of code sections.

36. (original) The system of claim 35 wherein the file system section receives a second plurality of updated code sections; and, wherein the patch library replaces the second plurality of code sections in the code storage section with the second plurality of updated code sections.
37. (original) The system of claim 34 wherein each symbol library comprises symbols having related functionality.
38. (original) The system of claim 34 further comprising: an airlink interface to receive new code sections; and, wherein the file system section stores the new code sections received via the airlink interface.
39. (original) The system of claim 38 wherein the code storage section includes a shared read-write code section with the read-write data arranged within for the first plurality of symbol libraries.
40. (original) The system of claim 39 wherein the file system section receives an updated read-write code section; wherein the patch library replaces the read-write code section in the code storage section with the updated read-write code section from the file system section; and, wherein the arrangement of the new read-write code section with the current code sections in the code storage section forms updated executable system software.
41. (original) The system of claim 40 wherein symbol libraries are arranged to start at the start of code sections; and, wherein the code storage section includes a first table code section with a code section address table cross-referencing code section identifiers with corresponding start addresses.
42. (original) The system of claim 41 wherein the symbol libraries include each symbol being offset from its respective code section start address; and, wherein the code storage section includes a second table code section with a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers.
43. (original) The system of claim 42 wherein the file system section receives an updated first table code section and an updated second table code section; wherein the patch library replaces the first and second table code sections in the code storage section with the updated first and second table code sections from the file system section; and, wherein the arrangement of the new first and second table code sections with the current code sections in the code storage section forms updated executable system software.

44. (original) The system of claim 43 wherein the patch library is a symbol library arranged in a patch manager code section of the code storage section.
45. (original) The system of claim 44 wherein the file system section receives an updated first code section; and, wherein the patch library over-writes the first code section in the code storage section with the updated code sections in the file system section.
46. (original) The system of claim 45 wherein the file system section receives an updated patch manager code section; wherein the patch library replaces the patch manager code section in the code storage section with the updated patch manager code section in the file system section; and, wherein the arrangement of the updated patch manager code section with the current code sections in the code storage section forms updated executable system software.
47. (original) The system of claim 46 wherein the patch manager code section includes read-write data, the code section address table, and the symbol offset address table; and, wherein receiving an updated patch manager code section includes receiving an updated symbol offset address table, updated code section address table, and updated read-write data.
48. (original) The system of claim 46 wherein the patch manager code section includes symbol accessor code to calculate the address of a sought symbol using a corresponding symbol identifier, and a corresponding code section identifier; and, the system further comprising: a first location in memory for storing the symbol accessor code address.
49. (original) The system of claim 48 wherein the symbol accessor code calculates the address of the sought symbol by accessing the code section address table and the symbol offset address table.
50. (original) The system of claim 49 wherein the file system section receives an updated patch manager code section with updated symbol accessor code. wherein the patch library replaces the patch manager code section in the code storage section with the updated patch manager code section; and, wherein the arrangement of the updated symbol accessor code in the updated patch manager code section with the current code sections in the code storage section forms updated executable system software.
51. (original) The system of claim 50 wherein the file system section receives an updated symbol accessor code address; and, wherein the patch library replaces the symbol accessor code address in the first location in memory with updated symbol accessor code address.

52. (original) The system of claim 51 wherein the patch manager code section includes the first location in memory.
53. (original) The system of claim 52 further comprising: a read-write volatile memory to accept the read-write data, the code section address table, the symbol offset address table, the patch library, symbol accessor code, and the symbol accessor code address from the patch manager code section for access during the execution of the system software.
54. (original) The system of claim 53 wherein the code storage section includes a second plurality of contiguously addressed memory blocks identified with the corresponding second plurality of code sections.
55. (original) The system of claim 54 wherein the code storage section includes code sections sized to accommodate the symbol libraries arranged within, and memory blocks sized to accommodate the corresponding code sections.
56. (original) The system of claim 55 wherein the code storage section includes code sections sized to accommodate sizes larger than the symbol libraries arranged within.
57. (original) In a wireless communications device, a software updating system, the system comprising: a code storage section memory including executable wireless device system software differentiated into a first plurality of symbol libraries arranged in a second plurality of code sections; an airlink interface to receive new code sections, including an updated patch manager code section; a file system section memory for storing new code sections received via the airlink interface; wherein the code storage section includes a patch manager code section with a patch library to replace code sections in the code storage section with updated code sections, the patch manager code section further including a code section address table, a symbol offset address table, a symbol accessor code, read-write data, and a symbol accessor code address; and, wherein the arrangement of the new code sections, including the updated patch manager code section, with current code sections in the code storage section forms updated executable system software.